Algorithms Final Exam

March 13, 2002, 10:00–13:00

Write your name and i.d. number on every page that you hand in. Do not use any books, notes, friends, cellular phones, etc.

1. Let S be a string over the finite alphabet Σ. (That is, $S \in \Sigma^*$.) An edit operation on S is either a deletion of a single character from S, or the insertion of a single character. Let S_1 and S_2 be two strings of length n over Σ. (That is, $S_1 \in \Sigma^n$, $S_2 \in \Sigma^n$.) Give an algorithm that computes their edit distance, and analyse its running time as a function of n and $|\Sigma|$. (The running time should be polynomial.)

   For example, the edit distance between "aabdc" and "abdac" is 2.

2. A matching in a graph is a set of edges such that every vertex touches at most one of these edges. A vertex cover is a set of vertices such that every edge touchs at least one of these vertices. (For every vertex cover, the vertices not in the vertex cover form an independent set.) Prove that in every bipartite graph, the number of edges in a maximum matching is equal to the number of vertices in a minimum vertex cover.

3. Assume that you are given a set of n bolts, each of different size, and a set of n nuts, each of different size. Assume moreover that every nut fits exactly one bolt (and vice versa). You need to match correctly every nut to its bolt. In order to do so you are allowed to perform a sequence of comparison operations of the following type: take one nut and one bolt and see if they fit together. The outcome of this operation is either "they fit", or "the nut is too big", or "the nut is too small". Note that you are not allowed to compare two nuts directly, or two bolts directly. (Of course, the relative sizes of two nuts can be inferred indirectly by comparing them to bolts. Likewise, the relative sizes of two bolts can be inferred by comparing them to nuts.)

   Give a randomized algorithm that matches all the nuts to their correct bolts. The expected number of comparison operations made by the algorithm should be $O(n \log n)$. Try also to determine the leading constant in the O notation for the expected number of comparisons.

4. A clique with 9 vertices in a graph with n vertices can be found by exhaustive search in time $O(\binom{n}{9}) = O(n^9)$. Here we assume that for any two vertices u and v, checking whether (u, v) is an edge in G takes unit time. Show how to find a clique of size 9 in time n^δ for some $\delta < 9$ (if such a clique exists).

Good luck.